Intelligent Multi-Task Allocation and Planning for Multiple Unmanned Surface Vehicles (USVs) Using Self-Organising Maps and Fast Marching Method

Yuanchang Liu[1], Rui Song[1], Richard Bucknall[1] and Xinyu Zhang[2]

1. *Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, UK*
2. *Laboratory of Marine Simulation and Control, Navigation College, Dalian Maritime University, Dalian, China, 116026*

Corresponding author: Rui Song. E-mail: r.song.11@ucl.ac.uk. Address: *Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, UK*

Abstract

As a result of the advances in autonomous navigation technology, ocean based operations with increasing levels of complexity can be undertaken using unmanned surface vehicles (USVs). Presently, the trend of developing USVs is to use multiple USVs as a fleet to carry out single or multiple tasks in a cooperative and coordinated manner. To further support such a deployment, a new intelligent multi-task allocation and path planning algorithm has been proposed in this paper based upon the self-organising map (SOM) and the fast marching method (FMM). To specifically address the two critical issues of energy consumption and communication range, a novel energy coordination scheme as well as a task prioritising method have been proposed to efficiently assign tasks to a multi-USV system. The algorithm has been verified and validated through a number of computer-based simulations and has been proven to work effectively in both simulated and practical maritime environments.

Keywords

Unmanned surface vehicle (USV); Task allocation; Path planning; Multi-vehicle system

1. Introduction

The research into autonomous vehicles has attracted increasing interest in the recent decade. Such vehicles have been widely utilised in different domains including air, ground and sea depending upon the specific vehicle's mode of operation. Through exploitation of autonomous vehicles, benefits such as minimised personal risks and increased operation efficiency can be realised, especially when deployed on high risk and challenging tasks. However, despite the similarity in being capable of autonomously executing challenging tasks in the environment of interest, different levels of autonomy exist across these platforms with Unmanned Aerial Vehicles (UAV) and Unmanned Ground Vehicles (UGV) being equipped with relatively higher degrees of autonomy than Unmanned Surface Vehicles (USV) which are mainly operated in the maritime domain to support complex marine tasks. Various reasons, including the lack of technology development and market support, have contributed to this circumstance; however, with increasing importance being attached to the need for more extensive exploration and managed exploitation of the oceans, it is imperative that more research needs to be conducted towards the development of USV control autonomy.

In order to increase the degree of autonomy of USVs, apart from the improvement of an individual vehicle's capability by updating either or both the hardware and software, another promising and cost effective approach is to develop and exploit the concept of shared autonomy, for when multiple USVs are being simultaneously deployed. By working in cooperation, compared with single USV operation, multiple USVs are able to service a wider mission area, provide improved operational efficiency and increase fault tolerance capability. Because of these benefits, the strategy of deploying multiple USVs in a coordinated manner has been

welcomed by many researchers with one of the most famous examples being NASA Jet Propulsion Laboratory's (JPL) attempt to employ a swam of USVs for undertaking challenging maritime missions. By using an advanced multi-vehicle control architecture, experiments were carried out to demonstrate the superior capability of a swarm of USVs in supporting a demanding harbour patrol mission, which requires the USVs to collaboratively monitor and track hostile vessels, inspect vessels to infer intent and trailing suspect vessels [25]. Similarly, Raboin et al. [21] investigated using the multi-USV system for patrolling and guarding an asset in an environment with hostile boats and civilian traffic. The cooperative behaviour of different USVs was achieved by using a contract-based task allocation strategy so that by iteratively predicting the behaviours of both the USVs and the other vessels only the most appropriate task can be assigned to the corresponding USV.

It also should be noted that the cooperation and coordination behaviour of multiple USVs can be enhanced with the support from other types of vehicle, such as autonomous underwater vehicles (AUVs), to further improve the autonomy level of marine vehicles and consequently better assist with the execution of maritime missions. For instance, in Matos et al. [20], the concept of deploying an AUV and a USV in a coordinated manner has been first proposed, where the AUV was used to collect ocean data at subsurface level with the USV being used to provide the position information in real-time. Such an AUV-USV fleet has also been adopted by Zolich et al. [33] to support ocean sampling missions. The core of this research was to use the USV as a communication relay to handle the communication between the AUV and a control centre through Wi-Fi channels such that a real-time data collection (at AUV level) and data processing (at control centre level) can be achieved. A larger coordination of unmanned vehicles for maritime data acquisition missions was also studied in Vasilijević et al. [24], where manned vessels, AUVs, USVs and UAVs were simultaneously deployed. In order to ensure the robust communication between different types of vehicles, a common network system architecture was proposed to allow the seamless data transmission and a situation awareness platform was also used to monitor the mission progress and react to any environmental changes.

With the clear advantages of deploying USVs in a multi-vehicle manner, to utilise such a strategy in a more efficient way a general autonomous navigation structure has been proposed in Liu et al. [18]. Within such a structure, three different layers including a task allocation layer (TAL), path planning layer (PPL) and task execution layer (TEL) work coherently with the aim of the TAL solving the multi-task allocation problem for a multi-USV system, in essence the allocation of specific tasks to each of the multiple USVs in conjunction with each particular vehicle's capability. It is worth noting that in Liu et al. [18], only the PPL and TEL layers were successfully achieved, and to complete this multi-USV autonomous navigation structure, the design of the TAL layer has been specifically addressed in this paper by developing a series of intelligent algorithms for multi-task allocation.

In general, the task allocation problem can be regarded as synonymous with the Travelling Salesman Problem (TSP) and similarly, the multi-task allocation for multi-USV can be summarised as the Multiple TSP (MTSP). The TSP has the feature of NP-hardness, and it could be solved using both *exact* and *heuristic* algorithms. As mentioned in Liu et al. [19], even though the *exact* algorithm is capable of providing accurate solutions, the extensive computational time required to execute such algorithms has prevented it from being used in high dimension. In contrast, the *heuristic* based algorithm can give a solution for any dimension with comparatively slight compromise within the results but greatly improved computational efficiency. This has promoted increased research effort towards applying various heuristic algorithms (the genetic algorithm, ant colony algorithm, and the neural network algorithm) in solving TSP and MTSP problems.

It should be noted that due to advances in artificial intelligence technology, especially in the neural network research, an increasing number of studies have cast light on the application of

neural networks for solving complex problems such as MTSP. Among them, the self-organising map (SOM) is of special interest due to its features of intuitive appeal, relative simplicity and promising performance [27]. SOM has not only been applied in solving TSP from a purely mathematical standpoint but has also been adopted to effectively provide an optimal task allocation strategy for autonomous vehicles in support of complicated missions. Such an adoption has brought sounding effects for UAV [11], AUV [12] and mobile robots [3] in achieving mission scheduling and coordination. However, very limited research has taken place into the investigation of using SOM to solve the task allocation problem for individual USVs, let alone the multi-task allocation for multi-USV systems.

To fill this gap and assist with the wider application of multi-USV systems in support of future coordinated operation within the marine environment, this paper has proposed a new intelligent multi-task allocation algorithm based upon SOM. This algorithm is able to optimally schedule multiple tasks for USVs in a practical ocean environment, within which the existence of non-traversable areas (such as islands), the dimensions of the area and the impaired communication channels have been largely mitigated by novel solutions, which consist of the main contributions of this paper. The tasks are then allocated to each USV, which use path planning algorithms to generate feasible trajectories for each vehicle to follow and subsequently execute tasks. The path planning algorithm used in this paper is based upon the fast marching method (FMM), and the performance of the proposed algorithm has been validated in several different computer-based simulations.

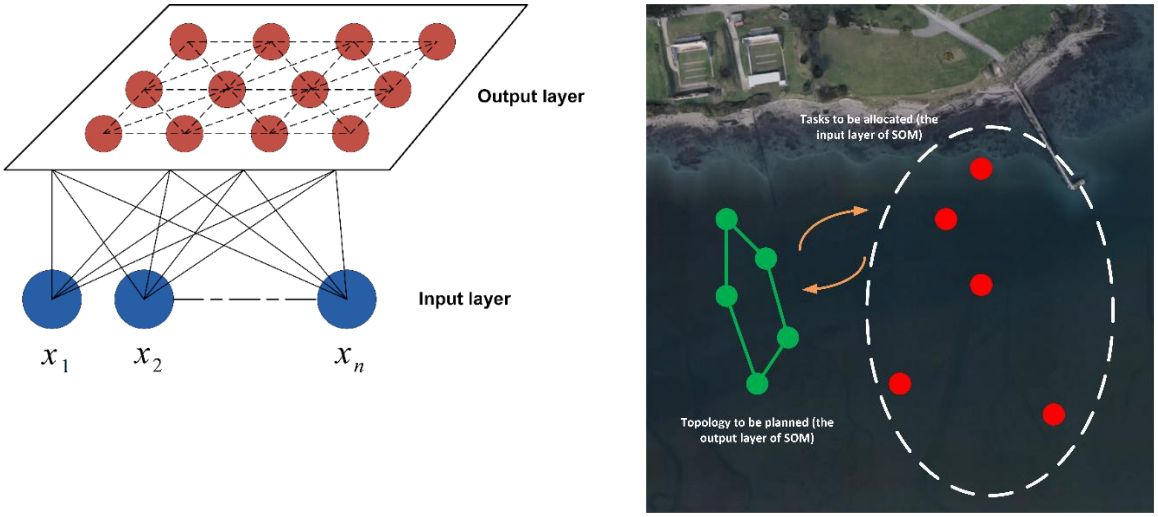
The reminder of this paper is organised as follows. Section 2 provides a literature review on using SOM to solve TSP/MTSP. Section 3 then specifically describes the proposed multi-task allocation algorithm of multi-USV system with Section 4 explaining how to use the FMM algorithm to generate trajectories and execute tasks. Section 5 provides a series of simulation results to validate and evaluate the proposed algorithm and Section 6 concludes the paper and discusses future work.

2. Related work

In this section, the fundamentals of the SOM will be introduced (in Section 2.1) then work in the field of using the SOM to solve multi-task allocation will be comprehensively reviewed (in Section 2.2). To give a thorough overview, rather than limiting the review to marine applications, autonomous vehicles in other domains have also been researched. Also, not only the TSP/MTSP based but some other interpretation of multi-task allocation problems has been reviewed as long as the SOM is applied as the main method. Finally, the fast marching method (FMM) will be briefly introduced in Section 2.3 with the details of the application of FMM on path planning problem being discussed in Section 4.1.

2.1. Self-organising map (SOM)

The self-organising map (SOM) proposed by Kohonen [16] is a type of artificial neural network that uses unsupervised learning to produce a low dimensional representation of an input space. The basic structure of the SOM is a two-layered network comprising an output layer and an input layer as represented in Figure 1(a). Neurons that need to be trained are contained in the output layer, and the training is achieved via the connections between the output and input layers. In general, the nodes on the input layer are randomly located while the output layer uses a two-dimensional regular topology such as a rectangular or hexagonal grid. Such a topology is largely maintained throughout the training process.



(a) (b)

Figure 1. Illustration of the self-organising map (SOM). (a) Basic structure of the SOM. A two-layered neural network including input layer and output layer is used. (b) In the application of the SOM for TSP, a circular topology for the output layer is used.

When applying the SOM to the TSP, each node in the input layer represents a city (or a task in a task-allocation problem) with a Cartesian coordinate (c_x, c_y) representing its location. The weights associated with the output layer neurons are in the same dimension and indicates the locations of neurons. Since the fundamental requirement of the TSP is to visit all the cities and eventually return to the start point, a circular topology (shown in Figure 1(b)) is thus used in the output layer. After forming such a topology, neurons then compete to become the winner throughout the training process, which adopts an unsupervised strategy consisting of two different procedures:

- **Winner selection**: a city (C_i) is first selected from the input space. The Euclidean distances between this city and all the neurons in the output layer are then calculated by:

$$d_{CW} = |C_i - W_j| = \sqrt{(C_{xi} - w_{xj})^2 + (C_{yi} - w_{yj})^2} \quad (1)$$

  where W_j represents the location of the j^{th} neuron, and the winner neuron is selected as the one having the minimum distance value:

$$W_{win} = \text{argmin}_W d_{CW} \quad (2)$$

- **Neighbourhood updating**: after determination of the winner neuron, the neighbourhood updating process will move the winner neuron, as well as its neighbour points towards new positions according to:

$$W_j = W_j + \mu * f(d, G) * (C_i - W_j) \quad (3)$$

  where μ is the learning rate, which determines the computation time. $f(d, G)$ is the neighbourhood function identifying the neighbourhood size of the winner neuron. The function can be viewed as a smoothing kernel which has the central influence on the winner neuron, and various forms can be used to define such a function with the most commonly used being:

$$f(d, G) = e^{-d^2/G^2} \quad (4)$$

  where d is the lattice distance on the topology and G is the gain parameter, which defines the width of the kernel (or the size of the neighbourhood). It should be noted that G should be a monotonically decreasing function of time. The rationale for this is during the initial stages, neurons are relatively far away from the cities, a large G the initial stages neurons are positioned relatively far away from the cities and a large G is

preferred to cause the neurons to move more quickly towards their corresponding city; whereas, at the latter stages, a stable situation has been formed and neurons become less competitive, therefore allowing a smaller value of G to be used.

2.2. Review of multi-task allocation using SOM

SOM was first used to solve the multi-task allocation problem for mobile robots in Zhu et al. [29]. By referring to the structure of SOM, input and output layers were redefined in a novel way so that the input layer represents the locations of different tasks and the output layer denotes the positions of robots. During each iteration of SOM, the updating process of the neural network is regarded as a combination of both task allocation and motion planning for the multi-robot system.

Based upon the work of Zhu et al. [29], work has continued to improve the algorithm and subsequently explore its application to other mobile platforms. Huang et al. [15] and Zhu et al. [30] employed the SOM method for dynamic task assignment and path planning for multi-AUV systems. To balance the equipped energy and the assigned workload, modification was made to the winner selection function (Equation (1)) in the conventional SOM so that only the tasks assessed to have the highest and realistic chances of achievement (tasks that can be executed by the AUV based upon available energy) will be assigned to an AUV. In addition, to better solve the problems that are normally encountered in actual maritime environments, such as the influence caused by variable ocean currents, a velocity synthesis approach was used to navigate the vehicle in favour of currents and consequently optimise the energy consumption. Zhu et al. [32] and Zhang et al. [27] further investigated the problem of obstacle avoidance and speed jumps (unrealistic speed change) of AUVs during the task allocation and path planning process when the SOM was used. The problems were resolved by integrating a new biologically inspired neural network, namely the Glasius bio-inspired neural network (GBNN), to represent the underwater environment such that tasks in the environment are assigned with attractive forces to attract the according AUV smoothly towards it while repulsive forces are applied for obstacles to help avoid collisions with such obstacles.

It should be noted that all of these works focused on solving a non-TSP based task allocation problem, i.e. vehicles are not required to start from a particular location and return to the same location on completion of the mission (a closed loop visit). This might be plausible for AUVs as an AUV can be launched and recovered from a mobile mother ship at any place, which makes the closed loop visit trivial. However, for the USV applications, the TSP like task allocation is important, as the vehicle is normally sent out from a fixed based station and needs to return to it when the mission is completed. Therefore, it becomes important to provide an insight on how the TSP based task allocation is solved using SOM in literature.

To solve the TSP based task allocation problem for autonomous vehicles, a series of works carried out by Faigl [5], Faigl et al. [6], Faigl [7] and Faigl [8] are representative. The main contribution derived from these studies is the adapting of a distance measurement metric that is used within the SOM updating process to make the SOM more suitable for autonomous vehicle applications, especially for solving the collision avoidance problem during navigation. It has been proposed that instead of using the Euclidean metric, a new metric that is an approximation of the shortest collision-free path between each neuron and the city will be used to find out the winner neuron. From the simulation results it is evident that all neurons can be kept outside the obstacle areas. Using such a new metric in the winner selection process for SOM, improvement works have also been done to solve the issues encountered in real applications. For example, Dubin's curve has been successfully integrated into the SOM to account for the vehicle's dynamic constraints [10]. An improved version of SOM, described as the growing SOM, was proposed in Faigl et al. [9] to provide a solution for autonomous data

collection from multiple sensors using an AUV. Specifically, to consider the cases where a large number of tasks have to be visited by an AUV within a certain time, tasks were prioritised according to sensors' capabilities and the proposed SOM was able to adaptively and optimally generate the neurons according to priority. Similarly, another sensor associated data collection problem was investigated in Best et al. [2] for a multiple robot system. The data collection problem was reformulated as to find a sensor visits sequencing problem, where each sensor was associated with several viewpoints that were determined by sensing ranges and self-occlusions. As each viewpoint has a unique but fixed reward value for each visit, the TSP problem was converted to a prize-collecting TSP, which could be efficiently resolved by employing the SOM algorithm.

Based upon the aforementioned works, studies of using SOM to specifically solve USV related problems were conducted by Liu et al. [19] and Song et al. [22]. In Liu et al. [19], a new collision avoidance approach was proposed by integrating an adaptive repulsive force field into the SOM. This has been mainly established upon the fact that when deploying USVs in the marine environment, the dimensions of obstacles can often vary with tides. Therefore, to take this into consideration and provide an effective collision avoidance strategy, the proposed adaptive repulsive force field is able to change in conjunction with the obstacle dimensions and consequently makes the new SOM applicable to different scenarios. Song et al. [22] used this new SOM algorithm on a practical USV for undertaking missions at the highlands of Peru. The on-board battery limit problem was specifically investigated by developing a new mission prioritising scheme.

Note that apart from using SOM in solving the multi-task allocation problem, evolutionary algorithms, such as the genetic algorithm (GA), ant colony optimisation algorithm (ACO) and particle swarm optimisation (PSO) algorithm, have also gained popularity and been successfully applied on different types of platforms including AUVs, USVs and UAVs. For example, Cai et al. [4] proposed to use the genetic algorithm to solve the multi-task allocation problem for a swarm of AUVs in a three dimensional (3D) underwater environment. The nonholonomic motion constraints of AUVs were specifically addressed by using interpolated 3D Dubin's curves that were compliant with vehicles' dynamic characteristics. Zadeh et al. [26] investigated AUV's task allocation and path planning by using a two-layered control architecture that consists of a global route planner based upon genetic algorithm and a local path planner using particle swarm optimisation algorithm. The change of the environmental factors, such as the change of the ocean currents, has been considered and addressed to promote the robustness and reactive ability of the proposed control architecture. Zhen et al. [28] proposed to an ant colony optimisation (ACO) algorithm based method for the search-attack mission planning for a multi-UAV platform. To solve the issue of maximum flight range that is constrained by the UAV's onboard battery limit, a state transition rule was incorporated into the algorithm to ensure that UAVs were able to fly back to their starting point when the assigned tasks were out of the reachable range. Although these research works can successfully resolve the problem of task allocation and path planning, the tasks are normally assessed and assigned based upon preliminary criteria such as the minimum travelling distance, and some sophisticated constraints, in particular the communication capability, have been ignored. Furthermore, if additional planning constraints are included, more optimisation costs have to be considered making the computational burden increase, which to some extent can reduce the computational speed of the evolutionary algorithms. However, such a problem would not prevent the application of SOM based algorithms as one of the main benefits of SOM is its capability of addressing multiple subproblems together and provide the solution simultaneously using the unsupervised learning [12].

In addition, by analysing and comparing these studies, it can be found that although a large number of studies have investigated the multi-task allocation problem on both single and

multiple vehicle platforms, there has been limited work on USV applications, not mention the multi-task allocation for multi-USV systems. The aim of this paper is therefore to investigate this area by using the SOM with particular interest in solving the issues that have mainly prevented USVs from being deployed in practical applications, including:

a) energy constraint multi-task allocation for the multi-USV system;
b) continued safe operation during periods of compromised communication capability between USV and the base station in marine environments;
c) collision avoidance in a confined environment with multiple obstacles;

2.3. Fast marching method (FMM)

The fast marching method (FMM) was first proposed by J. Sethian in 1996 as a numerical way to iteratively solve the Eikonal equation to simulate the propagation of an interface [23]. The Eikonal equation has the form:

$$|\nabla T(\boldsymbol{x})|V(\boldsymbol{x}) = 1 \tag{5}$$

where $T(\boldsymbol{x})$ is the interface arrival time at point $\boldsymbol{x}$ and $V(\boldsymbol{x})$ is the interface propagating speed. Equation (5) belongs to the partial deferential equation (PDE) and its numerical solution can be obtained via the upwind deferential method when using the FMM. The solving process of the FMM is similar to Dijkstra's method but in a continuous way. Suppose (x,y) is the point that T(x,y) needs to be solved. The neighbour of (x,y) is a point set containing four elements $(x + \Delta x, y)$, $(x - \Delta x, y)$, $(x, y + \Delta y)$,$(x, y - \Delta y)$. T(x,y) can be obtained by:

$$T_1 = \min(T_{(x-\Delta x,y)}, T_{(x+\Delta x,y)}) \tag{6}$$

$$T_2 = \min(T_{(x,y-\Delta y)}, T_{(x,y+\Delta y)}) \tag{7}$$

$$\left|\nabla T_{(x,y)}\right| = \sqrt{\left(\frac{T_{(x,y)} - T_1}{\Delta x}\right)^2 + \left(\frac{T_{(x,y)} - T_2}{\Delta y}\right)^2} \tag{8}$$

$$\left(\frac{T_{(x,y)} - T_1}{\Delta x}\right)^2 + \left(\frac{T_{(x,y)} - T_2}{\Delta y}\right)^2 = \frac{1}{\left(V_{(x,y)}\right)^2} \tag{9}$$

where Δx and Δy are the grid spacing in the x and y directions. The solution of Equation 9 is given by:

$$T_{(x,y)} = \begin{cases} T_1 + \dfrac{1}{V_{(x,y)}}, & if\ T_2 \geq T \geq T_1 \\ T_2 + \dfrac{1}{V_{(x,y)}}, & if\ T_1 \geq T \geq T_2 \\ Quadratic\ solution\ of\ Equation\ (9)\ \ if\ T > \max(T_1, T_2) \end{cases} \tag{10}$$

The pseudocode of FMM is described in Algorithm 1 by using an example of simulating an interface propagation process over a grid map. In the initialisation process, the algorithm first assigns all the grid points with the arrival time of infinity. The grid points are then grouped into three different categories, i.e. the *Far*, *Known* and *Trial* point sets. Such a categorisation method is similar to Dijisktra's and the specific meanings of each group are as follows:

- *Far*: contains grid points with undecided arrival time value (*T*). In the first-time step when running the FMM, all grid points except the start points belong to *Far*;

- *Known*: contains grid points with decided arrival time values (*T*). Such values will not be changed when the algorithm is executed;
- *Trial*: contains grid points with calculated arrival time values (*T*); however, values may be changed when the algorithm is running.

During each iteration of the algorithm, the point with the smallest *T* will be selected from the *Trial* set and added into the *Known* set. Then, for the selected point, all the neighbour points will be updated by using Equation (10) to have new arrival time values, and neighbour points located in the *Far* point set will then be moved into the *Trial* point set for the next iteration. The algorithm will be terminated when the *Trial* point set is empty.

Algorithm 1 Fast Marching Method algorithm

Require: configuration space (χ), start point (p_{start})

1: assign all the grid points in χ with the cost of Infinity ▷ $Initialisation\ process$
2: $T(p_{start}) \leftarrow 0$
3: $Far \leftarrow all\ grid\ points\ in\ \chi$
4: $Known \leftarrow all\ grid\ points\ with\ known\ cost$
5: **for** each adjacent point *a* of *Known* point **do**
6: $Trial \leftarrow a \cup Trial$
7: $T(a) = costUpdate(a)$ ▷ $Using\ Equation$ (10)
8: **end for**
9: **while** *Trial* is not empty **do** ▷ $Update\ process$
10: $p \leftarrow point\ with\ the\ lowest\ cost\ in\ Trial$
11: remove *p* from *Trial*
12: $Known \leftarrow p \cup Known$
13: **for** each neighbour point *a* of *p* **do**
14: $\tilde{T}(a) = costUpdate(a)$ ▷ $Using\ Equation$ (10)
15: if $\tilde{T}(a) < T(a)$ then
16: $T(a) \leftarrow \tilde{T}(a)$
17: **end if**
18: **if** $a \in Far$ **then**
19: remove *a* from *Far*
20: $Trial \leftarrow a \cup Trial$
21: **end if**
22: **end for**
23: **end while**
24: **return** T

3. Multi-task allocation for multi-USV systems

3.1. Improved SOM with collision avoidance capability

The Euclidean, distance based, winner selection process allows the SOM to determine the closest neuron to the city and subsequently update the winner neuron together with its neighbourhood points. However, such an updating process will produce compromised results, especially in a constrained environment, where a number of obstacles exist. To overcome this limitation, an improved SOM with collision avoidance capability has been proposed in Liu et al. [19].

In fact, the Euclidean metric based, neuron update process in conventional SOM can be explained from the potential field perspective. By analysing Equation (3), it can be shown that the second term on the right-hand side of the equation is acting as an attractive force, which attracts the neurons towards the selected city. Therefore, following this logic, Liu et al. [19] added a repulsive force and the neuron updating process will follow a new equation:

$$W_j = \begin{cases} W_j + \mu * f(d,G) * (\frac{(C_i - W_j)}{\|(C_i - W_j)\|} + \underset{f_{rep}}{\longrightarrow}), \ if\, d_{obs}(W_j) \le d_{min} \\ W_j + \alpha_{speed} * \mu * f(d,G) * \frac{(C_i - W_j)}{\|(C_i - W_j)\|}, \ otherwise \end{cases} \quad (11)$$

and $\underset{f_{rep}}{\longrightarrow}$ is calculated from:

$$\underset{f_{rep}}{\longrightarrow} = \boldsymbol{F}_{rep}(w_{xj}, w_{yj}) \quad (12)$$

where $d_{obs}(W_j)$ calculates the minimum distance to the obstacles from the position of W_j . d_{min} is a predefined minimum distance. $\boldsymbol{F_{rep}}$ (•) is the function returning the local repulsive force at W_j 's position (w_{xj}, w_{yj}) by referring to the repulsive force vector field $\boldsymbol{F_{rep}}$. The repulsive force vector field has been generated in a process consisting of two steps: 1) creating a repulsive potential field capable of implicitly reflecting the risk of obstacles. and 2) calculating the gradients of the potential field to get the corresponding force vector field. The details of the repulsive force vector field generating process can be referred to in Liu et al. [19]. From Equation (11), it shows that when $d_{obs}(W_j) \le d_{min}$, neurons are deemed to be too close to obstacles and the obstacle avoidance is thus triggered by updating neurons based upon both attractive and repulsive forces; whereas, if neurons stay in the safe areas, the conventional updating is applied but a new parameter ($\alpha_{speed} > 1$) introduced to achieve a fast convergence speed.

The improved SOM algorithm inherits the key structure of the conventional SOM with the main change being the adoption of a new updating equation to find the winner neuro when the network is being updated. As a consequence, the computational complexity of the improved SOM algorithm is the same to the conventional one, which in general is $O(t_f(dn + qn^2))$, where n is the number of neurons in a SOM, and each neuro is q-dimensional, the input data are d-dimensional and t_f is the learning steps [16]. Note that in this research because multi-USV systems are assumed to be operating in a two-dimensional (2D) environment, q and d both equal to 2, and t_f should be configured according specific application and in this paper, it is set up to be 1000 to give an optimised convergence rate.

3.2. Multi-task allocation for multiple USVs based upon improved SOM

Based upon the improved SOM, an algorithm can now be designed to solve the multi-task allocation problem for multi-USV systems. As mentioned earlier, issues of energy consumption and communication range are the two main fundamental constraints that restrict the operation of USVs in wide range deployments. To effectively tackle these problems, two additional novel improvements have been designed and implemented in the algorithm with the pseudocode shown at Algorithm 2.

First, when dealing with the communication range problem, as shown in Lines 2 to 4 of Algorithm 2, the issue is resolved by prioritising each city within an environment. The development of this solution was based upon Faigl et al. [9] and the pseudocode for assigning the priorities is shown in Algorithm 3. It is assumed that each USV will only have one base station to which it transmits the data it collects in real-time, and such a transmission is confined within a certain range, which is denoted as dis_{range}. A priority list ($priority_{list}$) will first be initialised of a size equal to the number of cities. Then, for each city, if the distance between

the city and base station is less than the communication range, it is considered that a successful transmission can be established and retained and a maximum priority value ($value_{max}$) will therefore be assigned to the denoted city. However, if such a distance is greater than the communication range, a minimum priority value ($value_{min}$) is given to the city indicating that the USV is too far away to successfully broadcast the information to the base station. Such a priority list will be used in the task allocation algorithm (Algorithm 2) to ensure that only the tasks within the communication range will be regarded as valid and therefore be taken into account when updating the SOM as shown in Line 8 of Algorithm 2. It should be noted that although the priority has only been quantified in terms of distance at this juncture, Algorithm 3 provides a general structure for resolving the communication problem for multi-USV systems. Additional factors that influence the communication capability can be integrated into the algorithm making it more suitable for implementation in practical applications.

As regards optimising energy consumption of multi-USV systems when allocating multiple tasks, this has been achieved by implementing a new coordinate updating scheme as shown in Line 7 in Algorithm 1. A new winner selection scheme is proposed using the equation:

$$W_{win} = \text{argmin}_W d_{CW\_Engergy} \tag{13}$$

where $d_{CW\_Engergy}$ is a new term measuring the distance between the city and neuron with the available energy storage been considered. $d_{CW\_Engergy}$ is calculated for each ring as:

$$d_{CW\_Engergy} = d_{CW} \cdot (1 + \frac{\sigma_{energy} \cdot length^i_{ring} - length_{ave}}{length_{ave}}) \tag{14}$$

where d_{CW} is the Euclidean distance between the city and the neuron on i^{th} ring. $length^i_{ring}$ is the length of the i^{th} ring, and $length_{ave}$ is the average length of all the rings. σ_{energy} is the energy adjustment parameter, which is used to ensure that winning neuron will be selected from the longest ring. Such a scheme has been established based upon the fact that if the USV is fuelled with sufficient energy, the vessel will be able to navigate a longer route and execute more tasks. Therefore, when updating the SOM, σ_{energy} is determined inversely proportional to the equipped energy storage onboard. For example, a smaller σ_{energy} value will be assigned to the ring that belongs to the USV having larger energy storage such that winner neuron can be more frequently selected from this ring and subsequently the updating process for other rings will be restricted.

The proposed multi-task allocation algorithm can specifically address the issues of limited energy consumption and constrained communication range, which are two main constraints for USVs operation in maritime environment and have not been properly considered in other works such as Faigl [8] and Best et al. [2]. Also, because of the implementation of the improved SOM as the base algorithm, compared with the algorithm in Faigl [8], the proposed algorithm has the feature of fast computational time, which makes the algorithm more suitable in large scale application where a USV swarm is deployed to conduct a large number of missions [19].

Algorithm 2 SOM based multi-task allocation for multi-USV algorithm

Require: set of input cities (χ) in two dimensional space, parameters of the improved SOM (d, G, μ, α_{speed}, d_{min}), the maximum number of iterations ($iter_{max}$), number of USVs (k), locations of base station (p_{base}), communication ranges (dis_{range}), maximum value of priority ($value_{max}$), maximum value of priority ($value_{min}$)

1: $R \leftarrow \{r_1, r_2, \ldots, r_k\}$ // Initialise *k* SOMs of *N* neurons with a ring topology for k USVs
2: **for** each base station **do**
3: $priority_i \leftarrow priorityDefine(\chi, p_{base}(i), dis_{range}(i), value_{max}, value_{min})$
4: **end for**
5: **while** iter $\neq iter_{max}$ **do**
6: randomly select a city (C_i) from χ
7: $W_i \leftarrow$ select the winning neuron of the city (C_i) from ring r_j ($j < k$) using Equation (7)
8: **if** *dis*(W_i, C_i) < $priority_j(C_i)$ **then**
9: **for** each neighbourhood point (W_i) of the winning neuron W_i **do**
10: update W_i using Equation ()
11: **end for**
12: **end if**
13: *iter* ← *iter* + 1
14: **end while**

Algorithm 3 Priority set-up algorithm (*priorityDefine* function)

Require: set of input cities (χ) in two dimensional space, the location of base station (p_{base}), communication range (dis_{range}), maximum value of priority ($value_{max}$), maximum value of priority ($value_{min}$)

1: $priority_{list} \leftarrow$ priority list for each city in the space with regard to the base station
2: **for** each city in the space **do**
3: $dis \leftarrow distance\ between\ j^{th} city\ (C_j)\ and\ the\ base\ station$
4: **if** $dis < dis_{range}$ **then**
5: $priority_{list}(C_j) \leftarrow value_{max}$
6: **else**
7: $priority_{list}(C_j) \leftarrow value_{min}$
8: **end if**
9: **end for**
10: **return** $priority_{list}$

4. Multi-goal path planning for multi-USV system

After each USV has been allocated its own set of tasks based upon the particular vessel's energy storage and communication capability, the next step is to make sure that each USV is able to execute these tasks by visiting them in sequence and to the most extent, appreciate the navigational safety and avoid any potential collision *en route*. To achieve this, path planning functionality will be called to generate feasible trajectory for each USV. Similar to work in Liu et al. [19], the trajectory has been calculated on the basis that the minimum distance cost should be maintained, i.e. the straight line between two points should be considered as an ideal candidate if the line does not violate any obstacle area. With the existence of obstacles, the ideal path (the straight line) should be accordingly and appropriately modified such that collision risks can be eliminated.

To implement such a concept, a two-step path planning strategy has been adopted in this paper for the multi-USV system. In the first step, for each USV off-line path planning is first used to

find the shortest path between each task by connecting each pair of mission points with a straight line. However, it should be noted that even though each task (mission point) can have one-to-one mapping to a neuron from a SOM after the updating process, redundant neurons still exist making the simple connecting of neurons with a straight line unfeasible in certain cases. As illustrated in Figure 2, it can be seen that in Area A, two neurons sitting on the straight line between two cities become unnecessary allowing them to be deleted and still retain the original path; whereas in Area B, as shown by the dashed line circled, neurons can not only be removed, but a new route can be generated by connecting two cities with a straight line. Therefore, by implementing the neuron deletion algorithm proposed in Angeniol et al. [1], each SOM ring can be reduced to a minim um number of neurons with each pair of neurons been connected by a straight line as the optimal path.

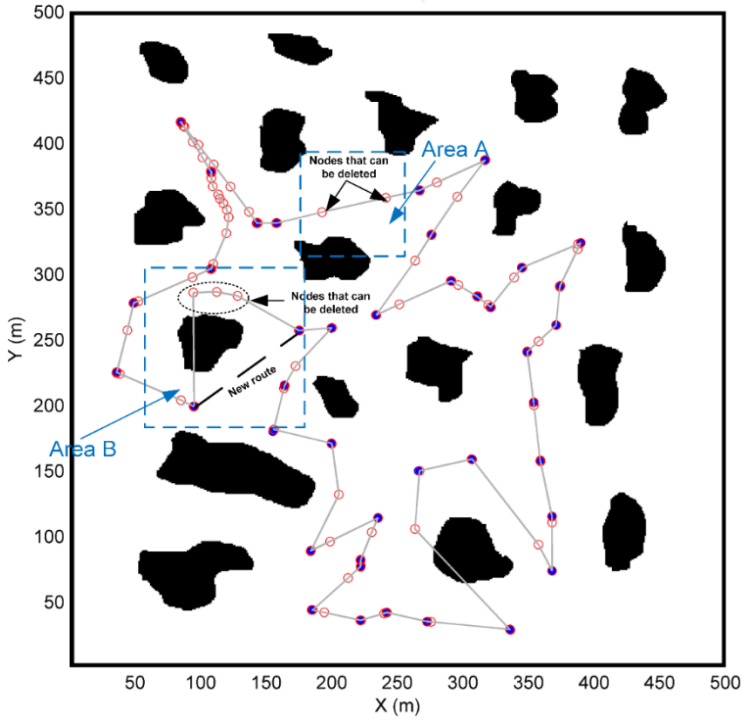


Figure 2. Node deletion process. Area A and B contain nodes that need to be removed.

Paths provided by the first step can be used as the off-line trajectories for the multi-USV system, and while each USV starts to track its individual path, some obstacles still require further path refinement to reduce collision risks. These collision risks could be the result of either already known obstacles where the connection between two missions may still pass through, or by some newly emerged obstacles that are detected by the vessel during the navigation. Therefore, to properly address these issues, a sophisticated on-line path planning algorithm should be used to regenerate the path in real-time. As there is a requirement that the path should be recalculated in an efficient way to ensure navigation performance will not be compromised, a path planning algorithm with fast computational time has to be selected. Among the most commonly used path planning algorithms, A* can provide the path with shortest distance but cannot guarantee the smoothness of the path. RRT related algorithms search for the path using the sampling based scheme but is more suitable in high-dimension space and returns the path relatively close to obstacles. The fast marching method (FMM) based algorithm can overcome these issues encountered by A* and RRT and therefore is used in this paper. In the following section, details of the FMM and its path planning algorithm will be provided.

4.1. Fast marching method based path planning

The FMM based path planning algorithm is described in Algorithm 3. Consider the planning space (***M***), to which the algorithm is to be applied, has a representation of a binary map and is perfectly rasterised. The algorithm first reads in ***M*** and calculates its according speed matrix (***V***). The speed matrix (***V***) has the same size as ***M*** and defines the interface propagation speed at each point in ***M***. Based on ***V***, the FMM is executed to calculate an arrival time matrix $\boldsymbol{T}_{FMM}$, and upon the time matrix $\boldsymbol{T}_{FMM}$, the optimal path is finally searched by applying the gradient descent method from the end point to the start **point.** Further explanation of FMM based path planning can be referred to in Garrido et al. [14].

Algorithm 4 FMM path planning algorithm

Require: planning space (***M***), start point (p_{start}), end point (p_{end})

1: Calculate speed matrix ***V*** from ***M***
2: $\boldsymbol{T}_{FMM} \leftarrow FMM(\boldsymbol{V}, p_{start})$
3: $path \leftarrow gradientDescent(\boldsymbol{T}_{FMM}, p_{start}, p_{end})$
4: **return** $path$

To demonstrate the effectiveness of the FMM based path planning algorithm, a comparison showing the difference between FMM and the other two classical approaches including A* and rapidly random tree (RRT) is shown in Figure 3. The comparison test was taken in a map with three randomly located obstacles. The dimension of the map is 700 pixels * 700 pixels and the start and end points were located at (100, 100) and (600, 600), respectively.

By observing the results in Figure 3, it can be summarised that paths generated by conventional algorithms such as RRT and A* (in Figure 3(c) and 3(d)) normally consist of several segments making them non-continuous. Hence, additional path smoothers are necessary to be applied to increase the continuity. In contrast, the FMM algorithm is able to provide the path with improved continuity and smoothness (in Figure 3(a)). In addition, as stated in Garrido et al. [14], two more additional benefits can be provided by FMM based path planning algorithm:

- *Completeness*: some path planning algorithms developed using the evolutionary searching algorithm (genetic algorithm, ant colony algorithm and etc.) suffer from the problem of searching incompleteness, which means the algorithm may fail to find a path in a complex environment. The algorithm developed based upon the FMM adopts a deterministic searching scheme ensuring a path can always be found as long as it exists.
- *Fast computation time*: the FMM algorithm is fast in dealing with the searching problem due to its low computation complexity. For a grid map having total grid number of *N*, the time complexity is *O(Nlog(N))* [14]. Such a feature is especially useful for practical path planning, where a fast decision making process is preferred.

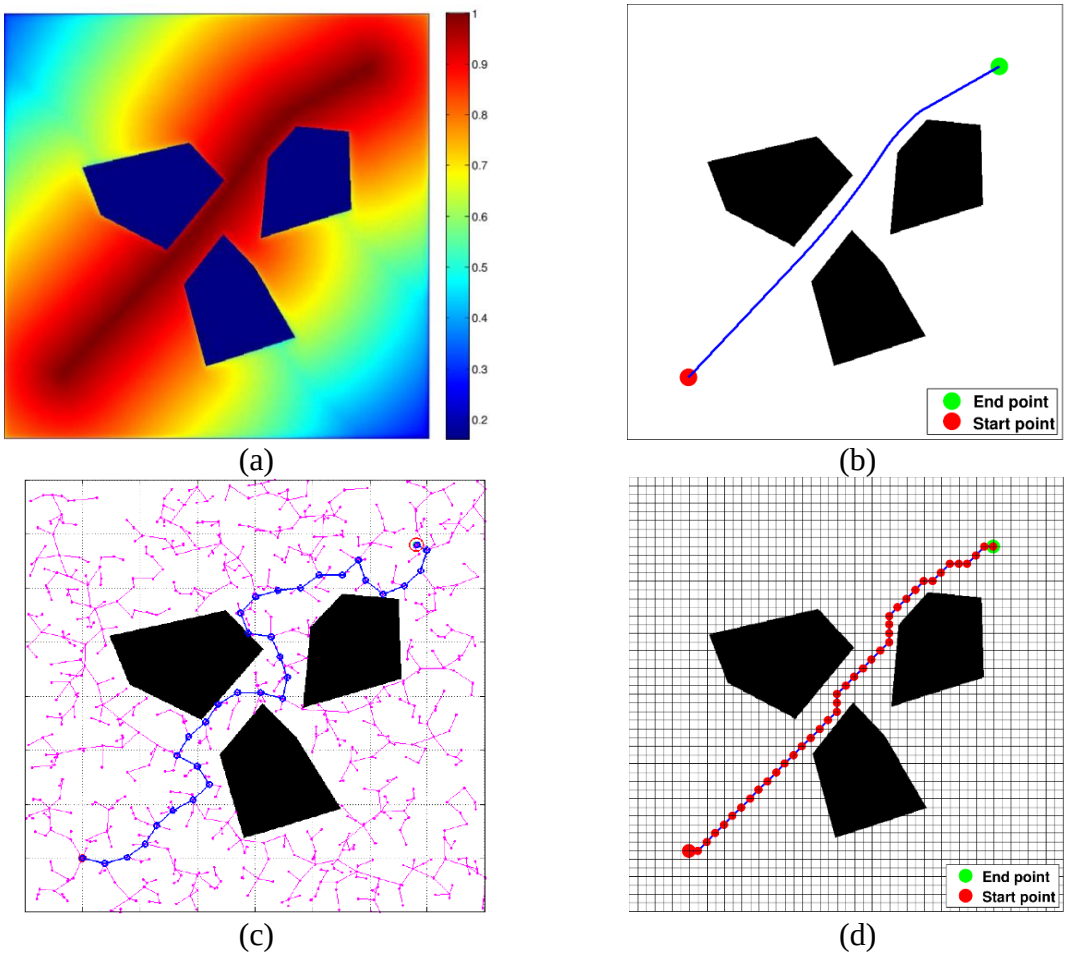


Figure 3. Comparison of FMM based path planning, A* and RRT algorithms in a congested environment including multiple obstacles. (a) Generated potential field when using the FMM algorithms. (b) Path generated using FMM based path planning algorithm. (c) Path generated using A* algorithm. (d) Path generated using the RRT algorithm.

5. Simulation results

To validate the effectiveness and performance of the proposed algorithm, a set of different computer-based simulations have been carried out. Tests can be generally categorised into two groups with the first (two simulations included) aimed at evaluating the task allocation capability for multi-USV system and the second (one simulation included) primarily demonstrating the algorithm performance in resolving both task allocation and path planning. In order to test the adaptability and robustness of the proposed algorithms, within the simulations, algorithm parameters are assigned different values according to specific mission requirements. Also, to facilitate the understanding of the simulation results and the effectiveness of the proposed algorithms, an overview of upcoming simulations can be briefly summarised as:

a) **Task allocation without regard to energy and communication constraints**: in this simulation, the core is to demonstrate how the SOM based task allocation algorithm can successfully allocate different tasks to a multi-USV system with the assumption been made that no energy or communication constraints will influence the multi-USV system. The updating process (or the training process) of the SOM neural network will be explicitly presented to reveal how the tasks are allocated in a balanced way.

b) **Task allocation under energy and communication constraints**: in this simulation, the primary aim is to test the algorithm's capability in dealing with energy and communication constraints using the proposed coordinating scheme as well as the task prioritising strategy. Different energy and communication constraints will be adopted to demonstrate that the proposed algorithm is robust enough to provide an optimised solution under all conditions.

c) **Multi-task allocation and path planning for multi-USV system:** in this simulation, multi-task allocation (SOM based) and path planning (FMM based) algorithms are integrated together and tested to validate that multiple tasks can be effectively assigned to USVs teams consisting of different number of vessels and a safe trajectory can always be efficiently generated while each vessel is executing tasks.

Algorithms have been coded in Matlab and simulations are run on the computer with a Pentium i7 3.4 GHz processor and 8 GB of RAM. A practical area, shown in Figure 4, has been used as the simulation environment, and has been converted to a 500 pixels * 500 pixels binary map that can be read and further processed by the algorithm. Within the environment, 70 water monitoring stations (shown as the yellow dots) are located in a random pattern and the water samples contained in the stations are due to be collected by a multi-USV system consisting of three USVs. These vehicles are required to establish a real-time communication channel to their corresponding base stations (marked as a star in green, magenta and red respectively). It should be noted that to maintain communication and data security, the communication channels here implement a one-to-one communication strategy where interfacing between channels is prohibited.

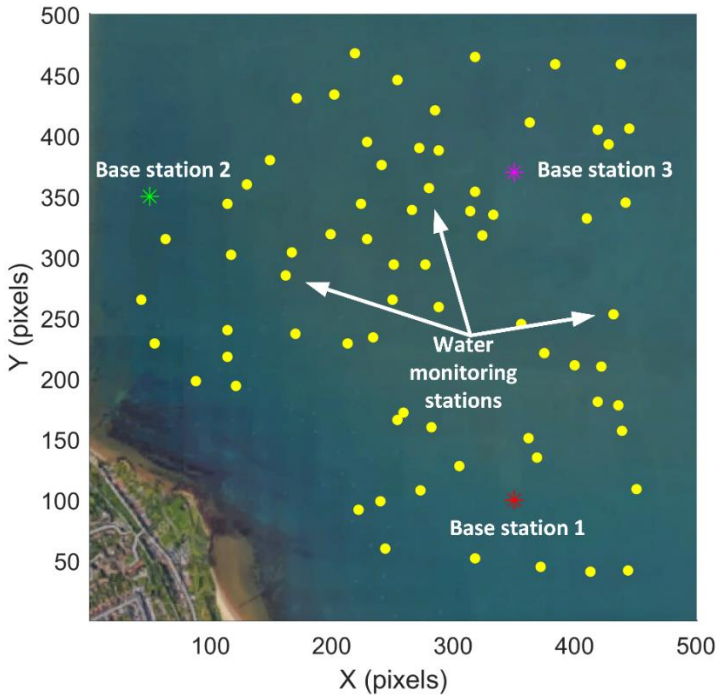


Figure 4. The simulation environment. Three base stations are shown in star shape and water monitoring stations are shown in yellow dots.

5.1. Simulation of multi-task allocation for multi-USV system

In this simulation, the main aim is to test the capability of the algorithm to assign 70 different tasks to three USVs subject to their amount of fuel on board (endurance) as well as the communication range.

5.1.1. Task allocation without regard to energy and communication constraints

First, the simulation has been carried out with the assumption that the three USVs have the same endurance, and because of this, the energy adjustment parameter (σ_{energy}) is set to the same value for each USV ($\sigma_{energy\_1} = 1$, $\sigma_{energy\_2} = 1$, $\sigma_{energy\_3} = 1$). In addition, it is also assumed that the communication range between each USV and its associated base station is greater than the largest dimension of the environment, so the communication range parameter (dis_{range}) in Algorithm 2 is assigned to be 750 pixels for each USV.

Simulation results are presented in Figure 5. The USV paths are planned such that they are each configured in a loop created by the SOM algorithm. Each path is initiated by a single neuron first being generated at the location close to the base station. Three rings are visually displayed in the colour of red, green and magenta representing the assigned tasks for USV 1, 2 and 3, respectively. When observing the updating process, each ring appears to be updated by the SOM in a way that after an equilibrium point is first reached, the ring will stay at this location and expand itself to reach as many tasks as possible. For example, the ring in green is first generated close to base station 2 (shown in Figure 5(a)). However, because this station is located at a position that is relatively distant from the tasks (shown as yellow dots), the ring will first move towards the bottom right area that is the site of more densely situated tasks, as shown in Figures 4(b)-(d). Once a state of equilibrium has been attained (as shown in Figure 5(d)), the algorithm will then seek to find the best matching task for each of the existing ring's neurons. The same process occurs to generate the other two rings (shown in red and magenta). However, as these two rings are initialised at base stations (1 and 3 respectively) that are in locations populated by a substantial number of tasks, it is not necessary for these two rings to be generated across a substantial distance to locate the nearest site that is best populated with tasks as the ring in green does in in its early stages of formation, and they can directly start the expanding process (shown in Figure 5(a)) and successfully find the matching task for all the neurons (shown in Figure 5(f)).

The process of determining the state of equilibrium during the SOM updating process can also be revealed when looking into the number of neurons generated. As stated earlier, an on-line neuron generation and deletion process is used to expand the SOM ring to make sure that an optimal number of neurons can be accessed when one-to-one mapping is formed between each neuron and its allocated task. This process can be observed when the number of neurons in each ring is recorded during each iteration. As seen from Figure 6, during initial stages, each ring has a relatively large number of neurons. However, after a transient period with variations of number of neurons exists for each ring, an equilibrium (or stable state) can be reached (at iteration around 275) for each ring with ring 1, ring 2 and ring 3 being assigned 23, 23 and 24 tasks respectively.

Another important aspect that should be noted is the algorithm's capability to assign the tasks based upon each USV's available energy. In this simulation, it is assumed that the three USVs each have an equal amount of energy and the energy adjustment parameter (σ_{energy}) is configured with the same value for each SOM ring. The result presented in Figure 5(f) clearly shows that the number of tasks assigned for each ring (USV) is roughly the same, which means the workload has been apportioned as equally as possible by using the proposed SOM algorithm. This is further supported by comparing the total distance covered by each USV

when executing these tasks. As can be seen from the 1st row in Table 1 that three USVs almost have the same distance cost with USV 1 required to cover 918 pixels, USV 2 for 849 pixels and USV 3 for 876 pixels.

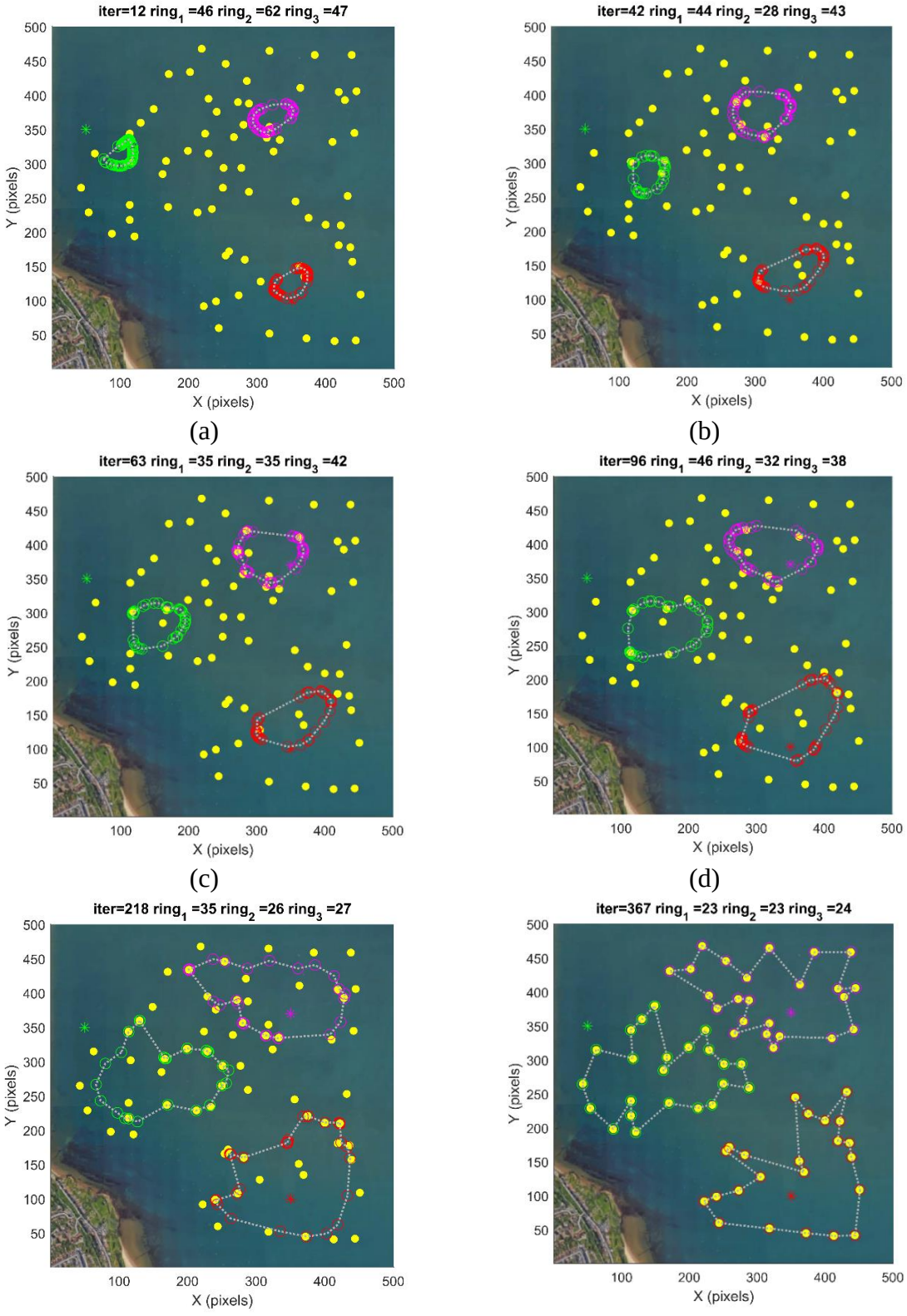


Figure 5. Simulation results of multi-task allocation for multi-USV system with equal amount of energy and no communication range constraints.

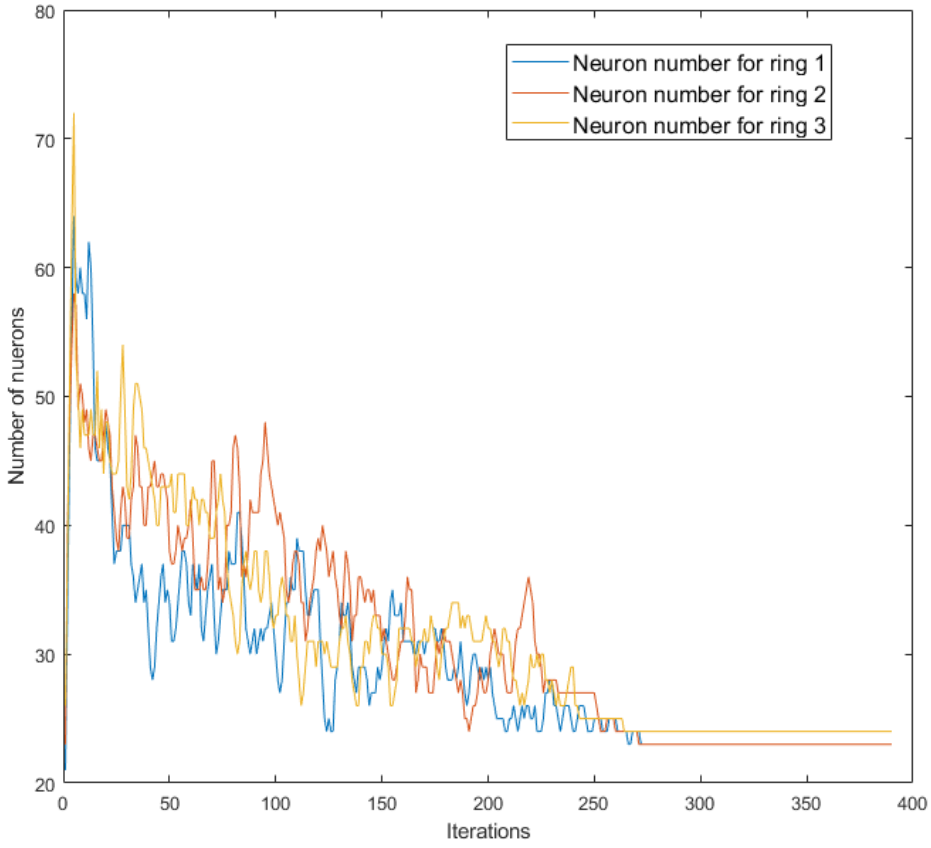


Figure 6. Number of neurons in each ring during each iteration.

Table 1. Comparison of total distance and number of tasks for each ring under different energy constraints.

	Ring 1		Ring 2		Ring 3	
	Total distance (pixels)	No. of tasks	Total distance (pixels)	No. of tasks	Total distance (pixels)	No. of tasks
$\sigma_{energy\_1} = 1$ $\sigma_{energy\_2} = 1$ $\sigma_{energy\_3} = 1$	918	23	849	23	876	24
$\sigma_{energy\_1} = 3$ $\sigma_{energy\_2} = 2$ $\sigma_{energy\_3} = 1$	588	14	682	17	1610	39
$\sigma_{energy\_1} = 1$ $\sigma_{energy\_2} = 5$ $\sigma_{energy\_3} = 1$	1192	28	255	7	1300	35
$\sigma_{energy\_1} = 8$ $\sigma_{energy\_2} = 2$ $\sigma_{energy\_3} = 1$	207	6	584	15	1986	49

5.1.2. Task allocation under energy and communication constraints

The algorithm is further tested to assess the functioning of the energy adjustment parameter (σ_{energy}). The main aim now is to validate the algorithm's capability of determining the task loading of each USV while considering each USV's energy availability.

In this simulation, using the same environment, USV 3 (magenta) is first assumed to have the highest amount of energy while USV 1 (red) has been allocated the lowest. The energy adjustment parameters have therefore been assigned as $\sigma_{energy\_1} = 3$, $\sigma_{energy\_2} = 2$, $\sigma_{energy\_3} = 1$ to make sure that during the SOM updating process USV 3 is able to maximise its energy consumption to explore the most tasks; whereas USV 1 has to reduce its number of tasks to save the energy. As the results in Figure 7(a) show the algorithm has registered the differences and has assigned 14, 17 and 39 tasks to USVs 1, 2 and 3 respectively. The total distance travelled by each USV further proves the effectiveness of the proposed algorithm. As shown in the second row of Table 1, the distances covered by USVs 1, 2 and 3 are 588 pixels, 682 pixels and 1610 pixels respectively, which stay well in accordance with the energy constraints applied to each USV.

When the energy constraints are changed to allocate USV with the lowest amount of available energy and USVs 1 and 3 with higher but nearly equal amounts of available energy, the task-allocation result is shown in Figure 7(b). It can be seen that the algorithm is also able to accordingly assign the tasks so that USV 2 will visit the least number of the tasks (7 tasks are assigned for the ring in green) to limit energy demand. To ensure the overall tasking can be completed, USVs 1 and 3 are allocated the remaining workload with USV 1 being assigned with 28 tasks and USV 3 with 35 tasks.

The algorithm's performance was also tested with more diverse boundary conditions. In this scenario, USV 1 has been designated as having very low reserves of energy compared to the other two, and the energy adjustment parameter (σ_{energy}) has been configured as $\sigma_{energy\_1} = 8$, $\sigma_{energy\_2} = 2$, $\sigma_{energy\_3} = 1$. From Figure 7(c) and the 4th row in Table 1, USV 1 has been assigned 6 tasks with the least distance of 207 pixels to cover. However, when the communication range constraint between a USV and its corresponding base station is applied, such a task-allocation result may become unfeasible. As shown in Figure 7(d) when a maximum 100 pixels communication range is imposed for USVs 1 and 2 (shown as the green and red circles respectively), it is clear that some tasks may not be attainable by USVs as they are outside the communication range making the vehicles incapable of transmitting the real-time information back to the base station.

The communication constraints are finally included by setting the communication range (dis_{range}) parameter to be 100 pixels in the proposed algorithm and the corresponding results are shown in Figure 8(a). As can be seen base stations 1 and 2 both have a 100 pixels communication range marked by the red and green circles respectively, a different task allocation strategy has been generated compared to that shown in Figure 7(d). For USV 2, all the tasks located within the communication range (tasks included within the green circle) can be allocated by fully mapping the neurons in the SOM to the tasks. Whereas for USV 1, most of the tasks (tasks included in the red circle) have been allocated expect one on the top left corner which is determined by the algorithm to be executed by USV 3. The reason for such a result is the algorithm now is not only subject to communication range constraints but also energy constraints, and as a result of USV 3 having the highest amount of energy and USV 1 having the lowest in this case, the algorithm intelligently allocates the workload between these two vehicles making USV 3 to handle more of the tasks. However, it also should be noted that as the equipped amount of energy is not able to provide enough endurance for USV 3 to cover

the remaining range of the tasking area, there are some tasks that are left unexplored uncompleted as shown in Figure 8(a). Another test result shown in Figure 8(b) can further demonstrate the capability of the proposed algorithm. In this case, all three base stations are subject to 100 pixels communication range constraints and the tasks allocated to the three USVs are retained within this range.

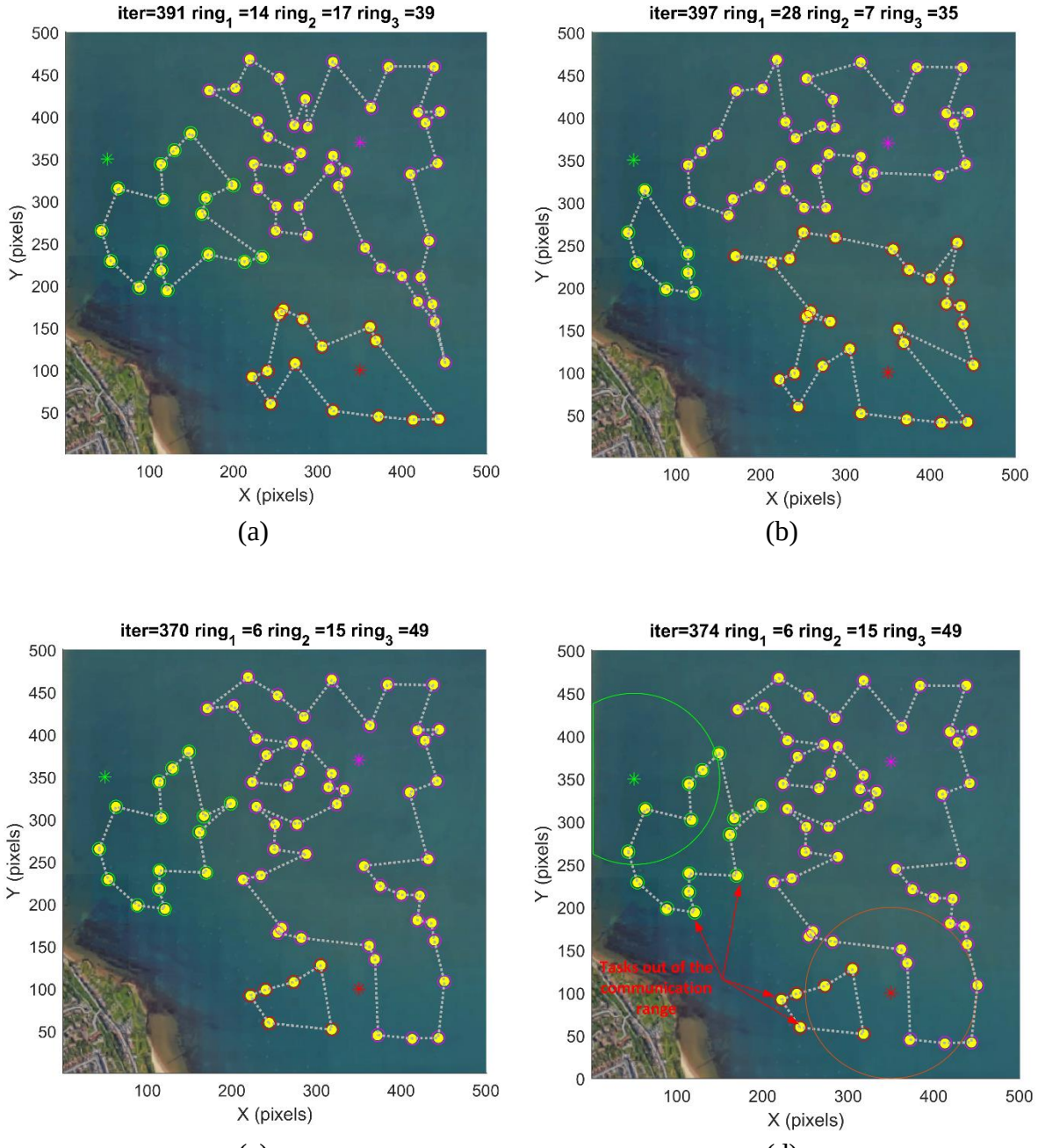
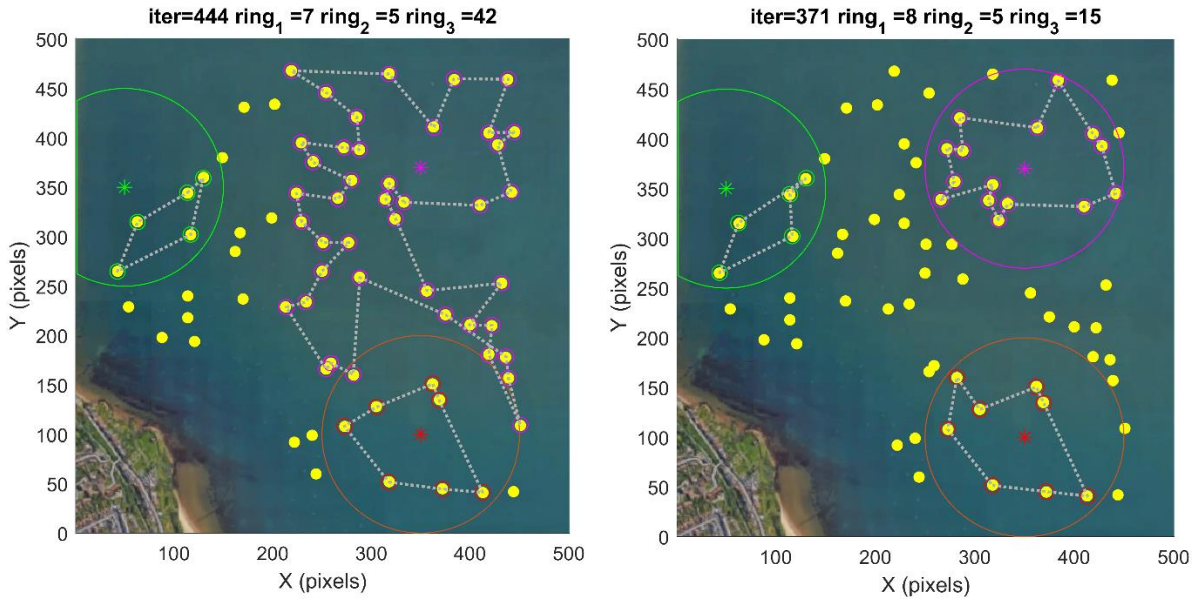


Figure 7. Simulation results of multi-task allocation for multi-USV system with different amount of energy. (a) $\sigma_{energy\_1} = 3$, $\sigma_{energy\_2} = 2$, $\sigma_{energy\_3} = 1$. (b) $\sigma_{energy\_1} = 1$, $\sigma_{energy\_2} = 5$, $\sigma_{energy\_3} = 1$. (c) $\sigma_{energy\_1} = 8$, $\sigma_{energy\_2} = 2$, $\sigma_{energy\_3} = 1$. (d) $\sigma_{energy\_1} = 3$, $\sigma_{energy\_2} = 2$, $\sigma_{energy\_3} = 1$. USV 1 and 2 have a 100 pixels communication range constraint.



(a) (b)

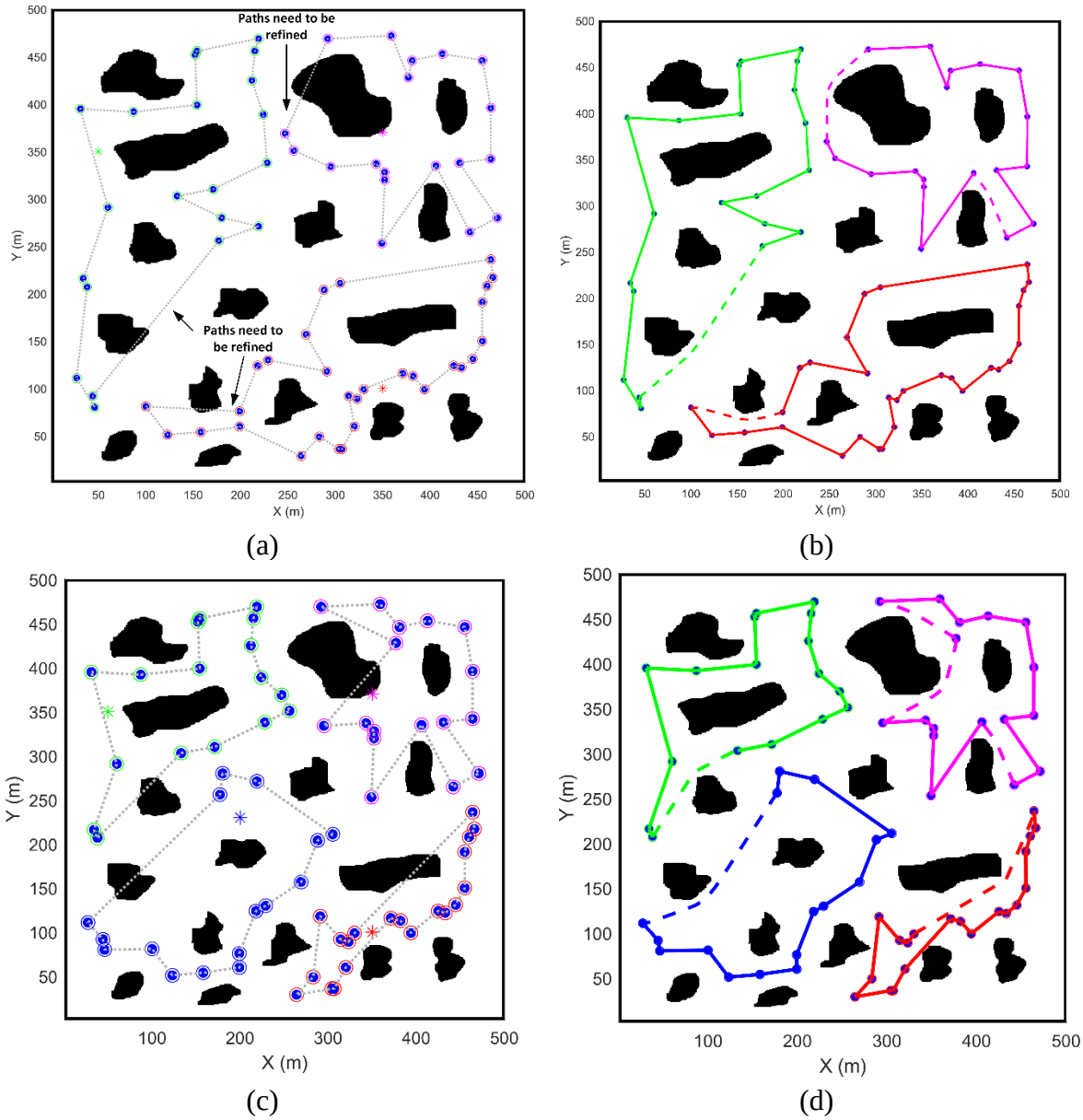
Figure 8. Simulation results of multi-task allocation for multi-USV system with equal amount of energy and different communication range constraints. (a) USV 1 and 2 have 100 pixels communication range and USV 3 has no communication constraint. (b) USV 1, 2 and 3 have 100 pixels communication range.

5.2. Simulation of multi-task allocation and path planning for multi-USV system

In this section, simulations have been undertaken to validate the proposed algorithm's capability of allocating multiple tasks to a multi-USV system and planning paths to execute said tasks. In order to rigorously test the collision avoidance capability, simulations have been carried out in an environment with multiple static obstacles (shown in Figure 9). The obstacles have irregular shapes and are randomly placed within the environment. A total number of 70 water monitoring stations are included and are scheduled to be visited by a multi-USV system. Within this test, in order to provide a thorough evaluation of the algorithm performance, multi-USV systems including 2, 3 and 4 vessels have been used. In addition, as the main purpose of these simulations is to demonstrate the effectiveness of the integration of task allocation and path planning as well as the collision avoidance capability, the energy and communication constraints are minor and the energy adjustment parameter (σ_{energy}) has therefore been set up with value of 1 for all USVs with no communication range constraints.

Simulation results are shown in Figure 9. In Figure 9(a) and (b), a case where the multi-USV system consists of three USVs has been tested. It clearly shows that the 70 tasks have been appropriately allocated to the three USVs, and most of the straight lines connecting the task pairs maintain a safe distance away from obstacles, which ensures that the USVs can safely follow these paths to execute tasks. However, as pointed out in Figure 9(a), some paths may violate the obstacle area making path refinement necessary. In Figure 9(b), the path recalculation process using the FMM algorithm has been presented. The recalculated paths shown as dashed lines provide a sound solution to ensure that any potential collision risks can be minimised. It also should be noted that because the main feature of the FMM algorithm is to provide a collision free path with the least distance cost, the total distance cost of the refined paths can be maintained to be minimum. Cases with different numbers of USVs are tested with Figure 9(c)-(d) and Figure 9(e)-(f) showing the results where 4 and 2 USVs are deployed, respectively. Clearly, any path that may lead to USV collision with an obstacle can be identified

by the algorithm, and its corresponding refined trajectory is provided as shown as the dash lines in Figure 9(d) and Figure 9(f).



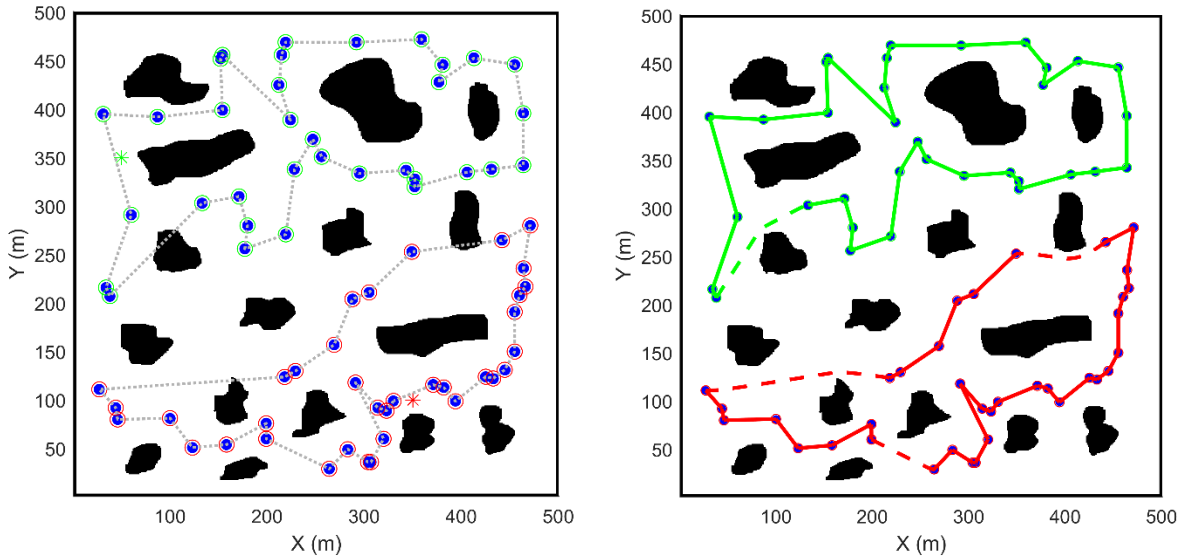


Figure 9. Simulation of multi-task allocation and path planning for multi-USV system. (a) Multi-task allocation for a multi-USV system with 3 vessels. (b) Path refinement for a multi-USV system with 3 vessels. (c) Multi-task allocation for a multi-USV system with 4 vessels. (d) Path refinement for a multi-USV system with 4 vessels. (a) Multi-task allocation for a multi-USV system with 2 vessels. (b) Path refinement for a multi-USV system with 2 vessels.

6. Conclusion and future work

In this paper, a new algorithm based upon the self-organising map (SOM) has been proposed to solve the problem of multi-task allocation for multi-USV systems. As the energy consumption is one of the main bottlenecks limiting the deployment of USVs for complex ocean tasks, a novel energy coordination scheme has been developed within the algorithm to balance the reserves of energy of each USV among the multi-USV system and proportionately assign tasks. In the meantime, a task prioritising algorithm has also been proposed to specifically take the communication range into account. By using such an algorithm, tasks can be allocated to make sure that the communication between each USV and its corresponding base station can be retained. After tasks are successfully allocated, a FMM based path planning algorithm is integrated to generate feasible paths for USVs for collision avoidance, which is the most important requirement in maritime navigation.

To further develop the algorithm, the energy coordination scheme can be improved with a more specified scheme revealing the relationship between actual energy consumption and the energy adjustment parameter (σ_{energy}) to be imposed. In the meantime, the communication constraint is currently only considered to be affected by distance. However, in reality, the strength of the communication channel could be influenced by other factors such as signal transmission power, signal noise power and signal-to-noise ratio [17]. Therefore, a more practical mathematical model of communication channel is worth investigation.

7. Acknowledgement

The authors are indebted to Mr. Konrad Yearwood for his valuable critique of this paper.

8. References

1) Angeniol, B., Vaubois, G. D. L. C., & Le Texier, J. Y. (1988). Self-organizing feature maps and the travelling salesman problem. *Neural Networks*, *1*(4), 289-293.
2) Best, G., Faigl, J., & Fitch, R. (2018). Online planning for multi-robot active perception with self-organising maps. *Autonomous Robots*, *42*(4), 715-738.
3) Best, G., Faigl, J., & Fitch, R. (2016, October). Multi-robot path planning for budgeted active perception with self-organising maps. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3164-3171). IEEE.
4) Cai, W., Zhang, M., & Zheng, Y. (2017). Task assignment and path planning for multiple autonomous underwater vehicles using 3D dubins curves. *Sensors*, *17*(7), 1607.
5) Faigl, J. (2010). Approximate solution of the multiple watchman routes problem with restricted visibility range. *IEEE transactions on neural networks*, *21*(10), 1668-1679.
6) Faigl, J., Kulich, M., Vonásek, V., & Přeučil, L. (2011). An application of the self-organizing map in the non-Euclidean Traveling Salesman Problem. *Neurocomputing*, *74*(5), 671-679.
7) Faigl, J. (2011). On the performance of self-organizing maps for the non-Euclidean Traveling Salesman Problem in the polygonal domain. *Information Sciences*, *181*(19), 4214-4229.
8) Faigl, J. (2016). An application of self-organizing map for multirobot multigoal path planning with minmax objective. *Computational intelligence and neuroscience*, *2016*.
9) Faigl, J., & Hollinger, G. A. (2018). Autonomous data collection using a self-organizing map. *IEEE transactions on neural networks and learning systems*, *29*(5), 1703-1715.
10) Faigl, J., & Váňa, P. (2016, September). Self-organizing map for the curvature-constrained traveling salesman problem. In *International Conference on Artificial Neural Networks* (pp. 497-505). Springer, Cham.
11) Faigl, J., & Váňa, P. (2017, May). Unsupervised learning for surveillance planning with team of aerial vehicles. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 4340-4347). IEEE.
12) Faigl, J., & Hollinger, G. A. (2018). Autonomous data collection using a self-organizing map. *IEEE transactions on neural networks and learning systems*, *29*(5), 1703-1715.
13) Faigl, J., & Hollinger, G. A. (2014, September). Unifying multi-goal path planning for autonomous data collection. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)* (pp. 2937-2942). IEEE.
14) Garrido, S., Moreno, L., Abderrahim, M., & Martin, F. (2006, October). Path planning for mobile robot navigation using voronoi diagram and fast marching. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2376-2381). IEEE.
15) Huang, H., Zhu, D., & Ding, F. (2014). Dynamic task assignment and path planning for multi-AUV system in variable ocean current environment. *Journal of Intelligent & Robotic Systems*, *74*(3-4), 999-1012.
16) Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464-1480.
17) Kim, S., Oh, H., Suk, J., & Tsourdos, A. (2014). Coordinated trajectory planning for efficient communication relay using multiple UAVs. *Control Engineering Practice*, *29*, 42-49.
18) Liu, Y., & Bucknall, R. (2015). Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Engineering*, *97*, 126-144.
19) Liu, Y., & Bucknall, R. (2018). Efficient multi-task allocation and path planning for unmanned surface vehicle in support of ocean operations. *Neurocomputing*, *275*, 1550-1566.
20) Matos, A., & Cruz, N. (2007, September). Coordinated operation of autonomous underwater and surface vehicles. In *OCEANS 2007* (pp. 1-6). IEEE.

21) Raboin, E., Švec, P., Nau, D. S., & Gupta, S. K. (2015). Model-predictive asset guarding by team of autonomous surface vehicles in environment with civilian boats. *Autonomous Robots*, *38*(3), 261-282.
22) Song, R., Liu, Y., Balbuena, J., Cuellar, F., & Bucknall, R. (2018, May). Developing an Energy Effective Autonomous USV for Undertaking Missions at the Highlands of Peru. In 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO) (pp. 1-7). IEEE
23) Sethian, J. A. (1996). A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, *93*(4), 1591-1595.
24) Vasilijević, A., Nađ, Đ., Mandić, F., Mišković, N., & Vukić, Z. (2017). Coordinated navigation of surface and underwater marine robotic vehicles for ocean sampling and environmental monitoring. *IEEE/ASME transactions on mechatronics*, *22*(3), 1174-1184.
25) Wolf, M. T., Rahmani, A., de la Croix, J.-P., Woodward, G., Hook, J. V., Brown, D., Schaffer, S., Lim, C., Bailey, P., Tepsuporn, S., Pomerantz, M., Nguyen, V., Sorice, C., and Sandoval, M. (2017). CARACaS multi-agent maritime autonomy for unmanned surface vehicles in the Swarm II harbor patrol demonstration. *Proceddings of Unmanned Systems Technology XIX*, *10195*, 1-11.
26) Zadeh, S. M., Powers, D. M., Sammut, K., & Yazdani, A. M. (2018). A novel versatile architecture for autonomous underwater vehicle's motion planning and task assignment. *Soft Computing*, *22*(5), 1687-1710.
27) Zhang, J., Feng, X., Zhou, B., & Ren, D. (2012). An overall-regional competitive self-organizing map neural network for the Euclidean traveling salesman problem. *Neurocomputing*, *89*, 1-11.
28) Zhen, Z., Xing, D., & Gao, C. (2018). Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm. Aerospace Science and Technology, *76*, 402-411.
29) Zhu, A., & Yang, S. X. (2006). A neural network approach to dynamic task assignment of multirobots. *IEEE transactions on neural networks*, *17*(5), 1278-1287.
30) Zhu, D., Huang, H., & Yang, S. X. (2013). Dynamic task assignment and path planning of multi-AUV system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace. *IEEE Transactions on Cybernetics*, *43*(2), 504-514.
31) Zhu, D., Liu, Y., & Sun, B. (2018). Task Assignment and Path Planning of a Multi-AUV System Based on a Glasius Bio-Inspired Self-Organising Map Algorithm. *The Journal of Navigation*, *71*(2), 482-496.
32) Zhu, D., Cao, X., Sun, B., & Luo, C. (2018). Biologically inspired self-organizing map applied to task assignment and path planning of an AUV system. *IEEE Transactions on Cognitive and Developmental Systems*, *10*(2), 304-313.
33) Zolich, A., Sœgrov, A., Vågsholm, E., Hovstein, V., & Johansen, T. A. (2017, May). Coordinated maritime missions of unmanned vehicles—Network architecture and performance analysis. In *2017 IEEE International Conference on Communications (ICC)* (pp. 1-7). IEEE.